

Supplementary Material for the article:
“Bayesian Cluster Weighted Gaussian Models”

Contents

A	Telescoping sampler	2
B	Technical details	3
C	Further simulation results	3
D	TCGA Dataset	6
D.1	Influential genes per cluster	6
D.2	Model diagnostics	6
E	Computing time	9
F	Competing models	9
F.1	Mixtures of regressions (FLEXMIX)	9
F.2	Mixtures of experts (FLEXMIX/MoEClust)	10
F.3	Flexible cluster weighted models (flexCWM)	10
F.4	Regularized joint mixture models (RJM)	11

A Telescoping sampler

Algorithm: Telescoping sampler for the BGCWM model

Iterate the following steps for $t = 1, \dots, T$:

0. Obtain starting values of latent allocations, parameters and hyper-parameters.

1. For $i = 1, \dots, n$, draw $\mathbf{z}_i \sim \text{Multinomial}(1, p_{i1}, \dots, p_{iK})$ with success probabilities

$$p_{ik} = \frac{\pi_k f_{N_1}(y_i | \alpha_k + \mathbf{x}_i^\top \boldsymbol{\beta}_k, \sigma_k^2) f_{N_p}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^K \pi_l f_{N_1}(y_i | \alpha_l + \mathbf{x}_i^\top \boldsymbol{\beta}_l, \sigma_l^2) f_{N_p}(\mathbf{x}_i | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}.$$

and permute the labels $\{1, \dots, K\}$ so that the first K_+ mixture components are not empty.

2. For steps (2.i)–(2.x) apply for $k = 1, \dots, K$:

2.i Set $n_k = \sum_i \mathbb{1}_{(z_{ik}=1)}$, $\mathbf{z}_k = (z_{1k}, \dots, z_{nk})^\top$ and $\mathbf{Z}_k = \text{diag}(\mathbf{z}_k)$.

2.ii Draw $\alpha_k \sim N(n_k^{-1} w_k (\mathbf{y} - \mathbf{X} \boldsymbol{\beta}_k)^\top \mathbf{z}_k, n_k^{-1} w_k \sigma_k^2)$, where $w_k = \sigma_\alpha^2 / (\sigma_\alpha^2 + n_k^{-1} \sigma_k^2)$.

2.iii Draw $\boldsymbol{\beta}_k \sim N_p(\mathbf{A}_k^{-1} \mathbf{X}^\top \mathbf{Z}_k (\mathbf{y} - \alpha_k \mathbf{1}_n, \sigma_k^2 \mathbf{A}_k^{-1}))$, where $\mathbf{A}_k = (\mathbf{X}^\top \mathbf{Z}_k \mathbf{X} + \mathbf{T}_k^{-1})$ and $\mathbf{T}_k = \text{diag}(\tau_{k1}^2, \dots, \tau_{kp}^2)$.

2.iv Draw $\sigma_k^2 \sim \text{IG}(a + \frac{1}{2}(n_k + p), b + \frac{1}{2}[\mathbf{e}_k^\top \mathbf{Z}_k \mathbf{e}_k + \boldsymbol{\beta}_k^\top \mathbf{T}_k^{-1} \boldsymbol{\beta}_k])$, where $\mathbf{e}_k = (\mathbf{y} - \alpha_k \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}_k)$.

2.v For $j = 1, \dots, p$, draw $t_{kj} \sim \text{InvGaussian}(\sigma_k \lambda_k \beta_{kj}^{-1}, \lambda_k^2)$ and set $\tau_{kj}^2 = t_{kj}^{-1}$.

2.vi Draw $\ell_k \sim \text{Gamma}(p + 0.5, 0.5(\sum_{j=1}^p \tau_{kj}^2 + \delta_k))$ and set $\lambda_k = \sqrt{\ell_k}$.

2.vii Draw $\delta_k \sim \text{Gamma}(1, 0.5(\lambda_k^2 + 1))$.

2.h Draw $\boldsymbol{\mu}_k \sim N_p((n_k + 1)^{-1}(\mathbf{m}_0 + \sum_i z_{ik} \mathbf{x}_i), (n_k + 1)^{-1} \boldsymbol{\Sigma}_k)$.

2.ix Set $\mathbf{S}_k = \sum_i z_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top + (\boldsymbol{\mu}_k - \mathbf{m}_0)(\boldsymbol{\mu}_k - \mathbf{m}_0)^\top$.

2.x For $j = 1, \dots, p$:

2.x.i Form the partitioned matrices $\boldsymbol{\Omega}_k^{(j)}, \mathbf{S}_k^{(j)}, \boldsymbol{\Phi}_k^{(j)}$ as defined in Eq. (26).

2.x.ii Draw $\eta_{k1} \sim \text{Gamma}(0.5(n_k + 1), 0.5(s_{kjj} + \psi_k))$ and $\boldsymbol{\eta}_{k2} \sim N_{p-1}(-\mathbf{C} \mathbf{s}_k^{(\setminus j, j)}, \mathbf{C})$,
where $\mathbf{C} = ((s_{kjj} + \psi_k) \boldsymbol{\Omega}_k^{(\setminus j, \setminus j)^{-1}} + \text{diag}(\boldsymbol{\phi}_k^{(\setminus j, j)})^{-1})$.

2.x.iii Set $\boldsymbol{\omega}_k^{(\setminus j, j)} = \boldsymbol{\eta}_{k2}$ and $\omega_{kjj} = \eta_{k1} + \boldsymbol{\eta}_{k2}^\top \boldsymbol{\Omega}_k^{(\setminus j, \setminus j)^{-1}} \boldsymbol{\eta}_{k2}$.

3. For $j < l$ ($l = 2, \dots, p$), draw $u_{kjl} \sim \text{InvGaussian}(\omega_{kjl}^{-1} \psi_k, \psi_k^2)$, set $\phi_{kjl} = u_{kjl}^{-1}$.

4. Draw $\psi_k \sim \text{Gamma}(r + 0.5p(p + 1), s + 0.5\|\boldsymbol{\Omega}_k\|_1)$.

5. Draw $\boldsymbol{\pi}(\mathbf{z}, \gamma, K) \sim \text{Dir}(\gamma_1 + n_1, \dots, \gamma_K + n_K)$, where $\gamma_k = \gamma/K$, $k = 1, \dots, K$.

6. Draw $K|(\mathbf{z}, \gamma)$ from Eq. (27)

7. Draw $\gamma|(\mathbf{z}, K, \nu_\ell, \nu_r)$ from Eq. (28).

Fixed parameters: $\sigma_\alpha^2 = 10^3, a = b = 10^{-2}, \mathbf{m}_0 = \mathbf{0}_p, r = 1, s = 10^{-2}, \nu_\ell = 6, \nu_r = 3, a_\lambda = 1, a_\pi = 4, b_\pi = 3$.

B Technical details

Initialization. The MCMC sampler for the BGCWM model as well as the EM algorithm in Perrakis et al. (2023) (denoted as RJM in what follows) are initialized by the same procedure. For this purpose, the initial values of latent allocation variables are set equal to the classification obtained by estimating a finite mixture of normal linear regression models as implemented in the `flexmix` package (Leisch, 2004; Grün and Leisch, 2007, 2008).

Specifically, for the case of fixed number of components, the BGCWM sampler was initialized by setting the allocation variables to the resulting single best clustering arising by running `flexmix` under 30 randomly chosen starting values. For the case of the overfitting mixture model or the telescoping sampler, we repeated the same initialization procedure for 10 randomly selected numbers of components in the set $6 \leq K \leq 15$. The final starting value is chosen as the one which corresponds to the highest log-posterior density $p(K, \gamma, \{\boldsymbol{\theta}_k, \pi_k, \sigma_k^2, \lambda_k, \psi_k\}_{k=1}^K | \mathbf{y}, \mathbf{X})$ (up to a normalizing constant) after an additional single iteration of the telescoping sampler.

Concerning the other methods in the comparisons of Section 4, `flexmix` was initialized after calling the `initFlexmix()` function under 100 random starts. The default initialization procedure was used in `MoEclust` and `flexCWM`.

MCMC details and post-processing. In our simulations in Section 4.1, the total number of MCMC iterations was set equal to $T = 11000$. The first 1000 iterations were discarded as burn-in period. Every 10th iteration was retained for inference. A larger number of MCMC iterations was considered in the real data application of Section 5. The retained MCMC samples were post-processed according to the Equivalence Classes Representatives (ECR) algorithm (Papastamoulis and Iliopoulos, 2010; Papastamoulis, 2016) in order to deal with possible label switching issues. We have also used the same method in order to suitably permute the final point estimates of the EM algorithm, and make direct comparisons against the true values as well as the between methods.

C Further simulation results

In this Section we provide more detailed insights to the results of the simulation study of Section 4.1. In particular, Figure C.1 displays the estimated number of clusters per simulation scenario. The corresponding graphs with the Adjusted Rand Index is given in Figure C.2.



Figure C.1: Estimation of the number of clusters (true values: $K \in \{2, 3, 4\}$) per covariate-simulation scenario.



Figure C.2: Adjusted Rand Index per covariate-simulation scenario.

D TCGA Dataset

In this section we provide further results for the analysis of the real dataset in Section 5 of the main manuscript.

D.1 Influential genes per cluster

Figure D.1 illustrates kernel density estimates of the marginal posterior distributions of the coefficients $\beta_{k,j}$ which correspond to variables picked up as significant according to the simultaneous 90% credible regions. Table D.1 lists influential genes (in parentheses the gene IDs) per cluster.

Cluster	Coefficient	Gene name and ID
1	$\beta_{1,3}$	WNK4 (65266)
1	$\beta_{1,4}$	SLC17A7 (57030)
1	$\beta_{1,5}$	LRRC25 (126364)
1	$\beta_{1,11}$	PI4K2B (55300)
1	$\beta_{1,13}$	CLDN10 (9071)
1	$\beta_{1,15}$	P2RY10 (27334)
2	$\beta_{2,6}$	SLFN1 (200172)
2	$\beta_{2,11}$	PI4K2B (55300)
3	$\beta_{3,6}$	SLFN1 (200172)
4	$\beta_{4,4}$	SLC17A7 (57030)
4	$\beta_{4,9}$	ARHGEF15 (22899)
4	$\beta_{4,13}$	CLDN10 (9071)
4	$\beta_{4,14}$	FAM54B (56181)

Table D.1: TCGA dataset: list of influential genes (in parentheses the gene IDs) per cluster.

D.2 Model diagnostics

To assess the fit of the model, we computed the classification residuals (see e.g. [Wadsworth et al. \(2018\)](#)). We first selected the most probable number of clusters from the telescoping sampler output, namely $K = 4$. The MCMC output was then post-processed to account for label switching, conditional on this choice of K . For each observation, we estimated its most likely cluster assignment $\hat{z}_i$ as well as the posterior means of the regression coefficients $(\hat{\alpha}_k, \hat{\beta}_k)$ and residual variance $\hat{\sigma}_k^2$, for each cluster $k = 1, \dots, 4$. The (standardized) classification residuals for cluster $\mathcal{C}_k := \{i = 1, \dots, n : \hat{z}_i = k\}$ are defined as

$$r_{i;k} = \frac{y_i - \hat{\alpha}_k - \sum_{j=1}^p \hat{\beta}_{kj} x_{ij}}{\hat{\sigma}_k}, \quad i \in \mathcal{C}_k.$$

The normal qq-plots of the classification residuals are displayed in Figure D.2, and they indicate that the model provides an adequate fit.

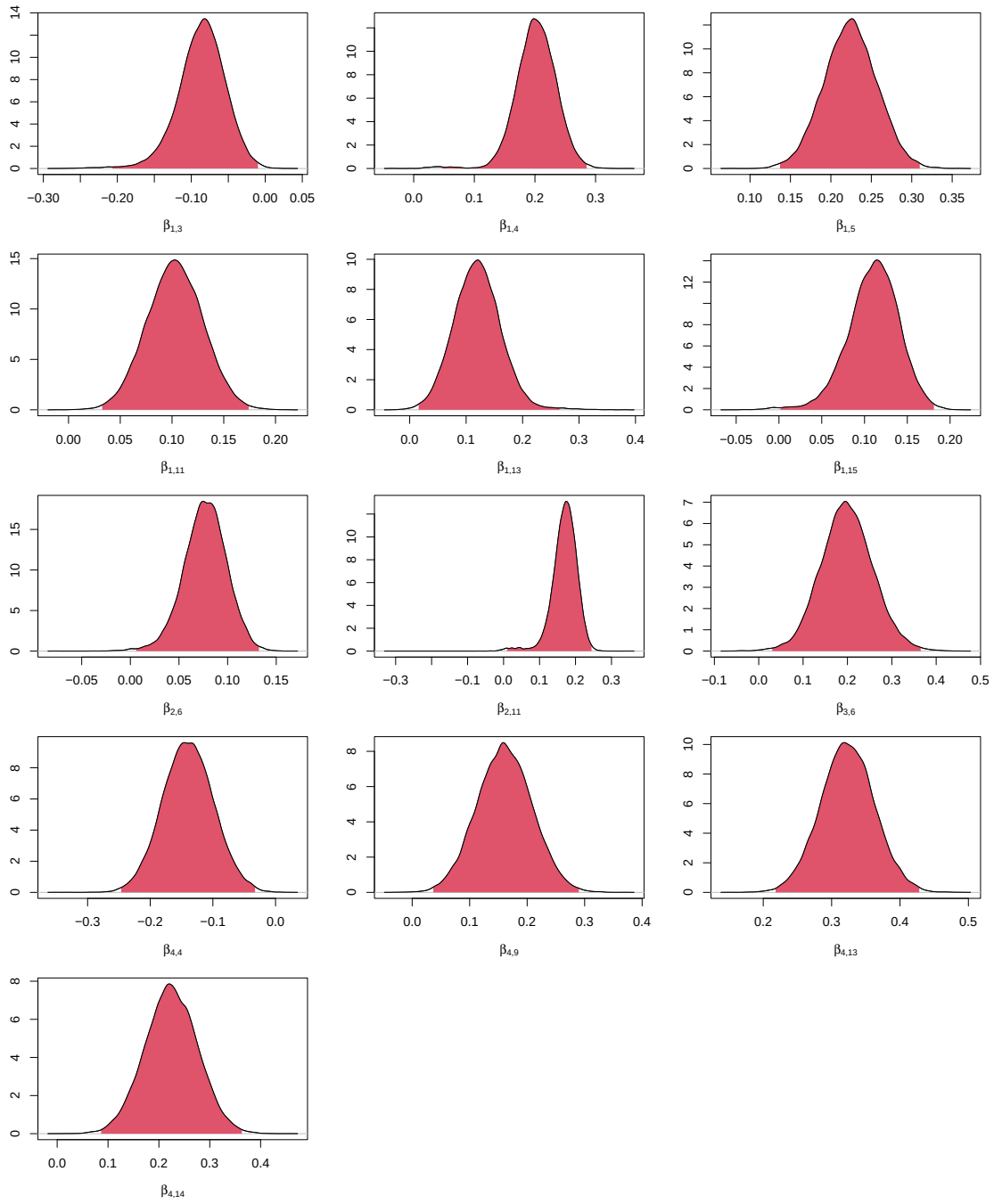


Figure D.1: TCGA dataset: Kernel density estimates of the marginal posterior distributions of the coefficients $\beta_{k,j}$ which correspond to variables picked up as significant according to the simultaneous 90% credible regions (shown in red).

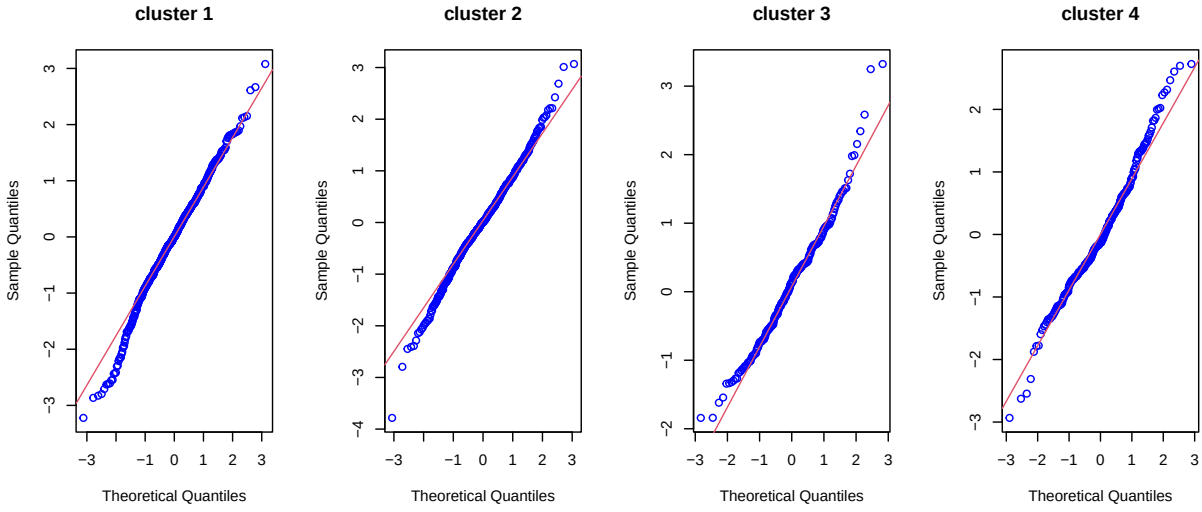


Figure D.2: Normal qq-plot of (standardized) classification residuals for the TCGA dataset arising from the output of the telescoping sampler, conditional on the most probable number of clusters ($K = 4$).

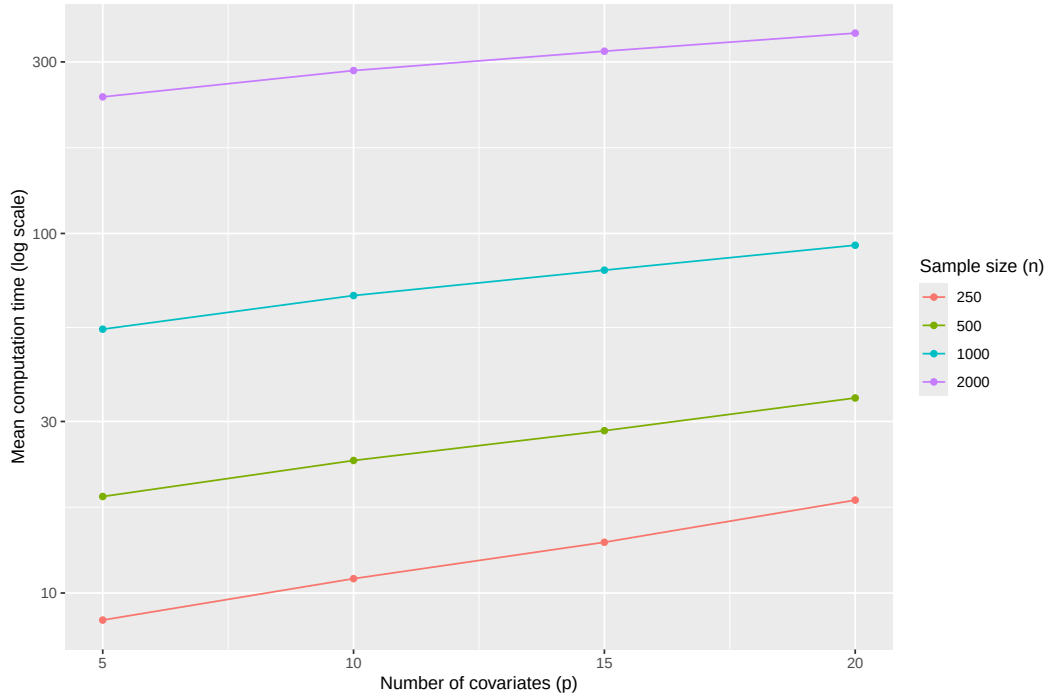


Figure D.3: CPU time (in seconds) for 1000 MCMC iterations. The y -axis is on $\log_{10}$ scale.

E Computing time

Figure D.3 illustrates the run-time for 1000 MCMC iterations of the proposed telescoping sampler for various values of the number of covariates (p) and sample size (n). The times are averaged over 10 repeated calls to the main function. All computations were carried out on a workstation running Ubuntu 24.04.3 LTS, equipped with an Intel Core i9 processor and 32 GB of RAM.

F Competing models

In this section, we provide more detailed descriptions of the underlying statistical models implemented in each software package considered in the main text (FLEXMIX, MoEClust, flexCWM, RJM).

F.1 Mixtures of regressions (FLEXMIX)

The observed likelihood for a finite mixture of regressions with K components is defined as

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\pi}, K) = \prod_{i=1}^n \sum_{k=1}^K \pi_k f(y_i | \boldsymbol{\theta}_k^y, \mathbf{x}_i), \quad (\text{F.1})$$

where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)^\top$ denotes the vector of mixing proportions with $\pi_k > 0$, $\sum_{k=1}^K \pi_k = 1$. The component densities $f(\cdot | \boldsymbol{\theta}_k^y, \mathbf{x}_i)$ belong to a (given) parametric family of distributions indexed by component-specific parameters $\boldsymbol{\theta}_k^y$, $k = 1, \dots, K$. The covariate vector $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^\top$, $i = 1, \dots, n$, is considered fixed. Finally, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k^y; k = 1, \dots, K\}$ denotes the vector of all component-specific parameters for $k = 1, \dots, K$. The implementation in the R package FLEXMIX (Leisch, 2004; Grün and Leisch, 2007, 2008; Gruen and Leisch, 2023) includes regression mixtures belonging to Gaussian, binomial/multinomial, Poisson and gamma families (among others).

For the particular case of finite mixtures of linear regressions (considered in our comparisons in Section 4), the (observed) likelihood is defined as

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\pi}, K) = \prod_{i=1}^n \sum_{k=1}^K \pi_k f_{N_1}(y_i | \alpha_k + \mathbf{x}_i^\top \boldsymbol{\beta}_k, \sigma_k^2), \quad (\text{F.2})$$

The component-specific parameters are $\boldsymbol{\theta} = \{(\alpha_k, \boldsymbol{\beta}_k, \sigma_k^2), k = 1, \dots, K\} \in \mathbb{R} \times \mathbb{R}^p \times (0, \infty)$, where α_k denotes the constant term of the linear predictor, $\boldsymbol{\beta}_k = (\beta_{k1}, \dots, \beta_{kp})^\top$ is the vector of regression coefficients for each explanatory variable and σ_k^2 denotes the variance of errors.

Parameter estimation for model (F.2) is carried out using the Expectation–Maximization (EM) algorithm for a range of values $K = 1, \dots, K_{\max}$. Ultimately, the number of components (K) is selected according to information criteria (such as BIC, AIC, or ICL).

The crucial difference with our model is in the way that we treat the covariates: mixtures of regressions consider that the covariates are fixed, while our cluster-weighted model treats the covariates as random.

F.2 Mixtures of experts (FLEXMIX/MoEClust)

The model defined in (F.2) can be further extended by allowing the mixing proportions $\boldsymbol{\pi}$ to depend on additional covariates, commonly referred to as *concomitant variables*. Accordingly, for observation $i = 1, \dots, n$, we denote the vector of mixing proportions by

$$\boldsymbol{\pi}_i = (\pi_{i1}, \dots, \pi_{iK})^\top,$$

where $\pi_{ik} > 0$ for $k = 1, \dots, K$ and $\sum_{k=1}^K \pi_{ik} = 1$. Thus, the (observed) likelihood of a mixture of experts with Gaussian components is defined as

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\pi}, K) = \prod_{i=1}^n \sum_{k=1}^K \pi_{ik} f_{N_1}(y_i | \alpha_k + \mathbf{x}_i^\top \boldsymbol{\beta}_k, \sigma_k^2). \quad (\text{F.3})$$

In general, the concomitant variables governing the mixing proportions $\boldsymbol{\pi}_i$ may differ from the explanatory variables entering the component densities f_{N_1} . In our comparisons in Section 4 of the main paper, however, the two coincide, and the mixing proportions depend on the same covariates $x_1, \dots, x_p$. That being said, the mixing proportions are modeled using a multinomial logit specification, given by

$$\pi_{ik} = \frac{e^{\dot{\alpha}_k + \mathbf{x}_i^\top \dot{\boldsymbol{\beta}}_k}}{\sum_{j=1}^K e^{\dot{\alpha}_j + \mathbf{x}_i^\top \dot{\boldsymbol{\beta}}_j}}, \quad k = 1, \dots, K$$

where $\dot{\alpha}_k \in \mathbb{R}$ and $\dot{\boldsymbol{\beta}}_k = (\dot{\beta}_{k1}, \dots, \dot{\beta}_{kp})^\top \in \mathbb{R}^p$ for $k > 1$. For identifiability, the first component is taken as the reference category, with $\dot{\alpha}_1 = 0$ and $\dot{\boldsymbol{\beta}}_1 = \mathbf{0}$. Alternative formulations are also available; for instance, Geweke and Keane (2007) proposes a multinomial probit specification.

Model estimation is carried out via the EM algorithm for $K = 1, \dots, K_{\max}$, and the number of components is selected using standard information criteria such as BIC, AIC, or ICL. In our empirical comparisons, finite mixtures of experts were estimated using the implementations in the R packages `FLEXMIX` (Leisch, 2004; Grün and Leisch, 2007, 2008; Gruen and Leisch, 2023) and `MoEClust` (Murphy and Murphy, 2020, 2023).

It should be clear that the main difference of this approach with our proposed model is the fact that we do not use covariate information in the mixing proportions and that we treat covariates as random.

F.3 Flexible cluster weighted models (flexCWM)

The family of cluster-weighted models considers the joint distribution of response variables $\mathbf{y}$ and covariates $\mathbf{X}$ by assuming a mixtures of regressions with random covariates, leading to an observed likelihood of the form

$$p(\mathbf{y}, \mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\pi}, K) = \prod_{i=1}^n \sum_{k=1}^K \pi_k f(y_i | \boldsymbol{\theta}_k^Y, \mathbf{x}_i) f(\mathbf{x}_i | \boldsymbol{\theta}_k^X), \quad (\text{F.4})$$

with $\boldsymbol{\theta} = \{(\boldsymbol{\theta}_k^Y, \boldsymbol{\theta}_k^X); k = 1, \dots, K\}$, where $\boldsymbol{\theta}_k^X$ and $\boldsymbol{\theta}_k^Y$ corresponding to the component-specific parameters, respectively, for $k = 1, \dots, K$. In the implementation of [Mazza et al. \(2018\)](#) (`flexCWM` package), the covariates can be of mixed-type, while the supported distributions are multivariate normal, binomial, Poisson, and multinomial. The estimation is performed within a maximum likelihood approach by means of the Expectation-Maximization algorithm.

In our setup, the distributions of the response variable and of the covariates are univariate normal and multivariate normal, respectively, that is

$$p(\mathbf{y}, \mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\pi}, K) = \prod_{i=1}^n \sum_{k=1}^K \pi_k f_{N_1}(y_i | \alpha_k + \mathbf{x}_i^\top \boldsymbol{\beta}_k, \sigma_k^2) f_{N_p}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (\text{F.5})$$

which reduces to Equation (5) of the main paper. Furthermore, the `flexCWM` package, considers the eigendecomposition of the covariance matrices $\boldsymbol{\Sigma}_k$, that is,

$$\boldsymbol{\Sigma}_k = \lambda_k \boldsymbol{\Gamma}_k \boldsymbol{\Delta}_k \boldsymbol{\Gamma}_k^\top, \quad k = 1, \dots, K \quad (\text{F.6})$$

where $\lambda_k = |\boldsymbol{\Sigma}_k|^{1/p}$, $\boldsymbol{\Delta}_k$ is the scaled (that is, $|\boldsymbol{\Delta}_k| = 1$) diagonal matrix of the eigenvalues of $\boldsymbol{\Sigma}_k$ sorted in decreasing order, and $\boldsymbol{\Gamma}_k$ is a $p \times p$ orthogonal matrix whose columns are the normalized eigenvectors of $\boldsymbol{\Sigma}_k$ ([Mazza et al., 2018](#)). Inspired by the work of [Celeux and Govaert \(1995\)](#), the `flexCWM` package considers a family of 14 parsimonious parameterizations for the K covariance matrices in Equation (F.6). These models, summarized in Table 1 of [Mazza et al. \(2018\)](#), are obtained by imposing equality constraints across components on the parameters λ_k , $\boldsymbol{\Gamma}_k$, and $\boldsymbol{\Delta}_k$, for $k = 1, \dots, K$. Each parameterization is fitted for $K = 1, \dots, K_{\max}$ and the optimal one is selected via information criteria.

Evidently, this family of cluster-weighted models enforces sparsity through parsimonious parameterizations of the covariance matrices of the joint distribution of the random covariates. In contrast, our Bayesian approach promotes sparsity by imposing a graphical-lasso shrinkage prior on the covariance matrices (Equation (22)). Moreover, sparsity in the regression coefficients is encouraged via a double-exponential shrinkage prior (Equation (15)). By embedding the model within a telescoping sampling scheme that jointly infers the number of components and clusters, we are able to fully quantify uncertainty in the clustering of observations in a Bayesian framework, rather than fitting multiple models for fixed values and then selecting a single model based on information criteria.

F.4 Regularized joint mixture models (RJM)

These models, introduced in ([Perrakis et al., 2023](#)), belong to the cluster-weighted family, so the observed likelihood is again

$$p(\mathbf{y}, \mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\pi}, K) = \prod_{i=1}^n \sum_{k=1}^K \pi_k f(y_i | \boldsymbol{\theta}_k^Y, \mathbf{x}_i) f(\mathbf{x}_i | \boldsymbol{\theta}_k^X). \quad (\text{F.7})$$

The approach in [Perrakis et al. \(2023\)](#) covers the case of univariate normal responses and of multivariate normal covariate matrices, so essentially, this leads to the familiar observed likelihood as

Equation (5) of the main paper:

$$p(\mathbf{y}, \mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\pi}, K) = \prod_{i=1}^n \sum_{k=1}^K \pi_k f_{N_1}(y_i | \alpha_k + \mathbf{x}_i^\top \boldsymbol{\beta}_k, \sigma_k^2) f_{N_p}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad (\text{F.8})$$

Estimation is handled via an Expectation-Conditional-Maximization (ECM) algorithm (R package `regjmix`, available at <https://github.com/k-perrakis/regjmix>). The approach considers potential determination of number of clusters based on information criteria (AIC and BIC).

What separates RJMs from the CWM literature is that here the authors consider regularization for the main parameters of interest via the assignment of prior distributions. Specifically, for the parameters of the normal linear regression part, we have the following prior distributions.

$$p(\alpha_k | K) \propto 1 \quad (\text{F.9})$$

$$p(\boldsymbol{\beta}_k | \sigma_k^2, \lambda_k, K) = \prod_{j=1}^p \frac{\lambda_k}{2\sigma_k} \exp \left(-\lambda_k \frac{|\beta_{kj}|}{\sigma_k} \right) \quad (\text{F.10})$$

$$p(\sigma_k^2 | K) \propto \frac{1}{\sigma_k^2} \quad (\text{F.11})$$

Note that the prior in (F.10) for the regression coefficients is the same as the one in Equation (15) in the main paper, but the priors for the intercepts (F.9) and the variances (F.11) are different (these are flat non-informative priors). The authors consider two different approaches for tuning the penalties λ_k ; one based on minimizing the cross-validation mean squared error and one based on placing a Pareto hyper-prior on each λ_k . It is worth noting that an additional, alternative prior, is considered for the regression coefficients; namely, the Normal-Jeffreys prior [Figueiredo \(2001\)](#), so that

$$p(\boldsymbol{\beta}_k) \propto \prod_{j=1}^p |\beta_{kj}|^{-1}. \quad (\text{F.12})$$

For the parameters of the multivariate distribution of the covariates, the RJM model considers regularization for the inverse-covariances $\boldsymbol{\Omega}_k = \boldsymbol{\Sigma}_k^{-1}$ of each group, using graphical-lasso priors of the form

$$p(\boldsymbol{\Omega}_k | \psi, K) \propto \left[\prod_{j=1}^p f_{\text{Exp}}(\omega_{kjj} | \psi/2) \prod_{j < l, l=2}^p f_{\text{DE}}(\omega_{kjl} | 0, \psi) \right] \mathbb{1}_{(\boldsymbol{\Omega}_k \in M^+)}, \quad (\text{F.13})$$

where $\psi > 0$ is the graphical lasso penalty, $\mathbb{1}_{(\cdot)}$ is the indicator function and M^+ is the space of positive definite matrices. Notice the similarity of the prior in (F.13) with the prior in Equation (22) in the main paper. The difference (which is important) is that in (F.13) the penalty parameter ψ is common in all groups and is actually fixed to a “universal” threshold given by $\tilde{\psi} = \sqrt{2n \log p}/2$ (discussed in [Städler and Mukherjee, 2013](#)). Interested readers are referred to [Perrakis et al. \(2023\)](#) paper for further details.

To conclude, let us highlight the main differences between RJMs and the proposed Bayesian CWM. First, apart from the double exponential in (F.10) the two models have different prior distributions. Not only that, but under the Bayesian CWM we add further levels of hierarchy

using the half-Cauchy distribution in Equation (14) for the λ_k 's and the gamma distribution in Equation (25) for the ψ_k 's. This allows us to pursue a fully Bayesian approach without having to rely on tuning procedures or “universal” thresholds. Secondly, the selection of number of clusters under RJMs is based on information criteria, while we develop a fully Bayesian approach through telescoping sampling (of course, we also examine the use of information criteria). Third, the estimation in Perrakis et al. (2023) is not fully Bayesian as it is based on the ECM algorithm which, essentially, delivers maximum-a-posteriori point estimates. Granted, this makes RJMs more scalable, on the other hand it does not provide full uncertainty quantification as the MCMC-based approach (it is worth considering that even finding confidence intervals is not straightforward using EMs).

References

- Celeux, G. and G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition* 28(5), 781–793.
- Figueiredo, M. A. T. (2001). Wavelet-based image estimation: An empirical Bayes approach using Jeffreys’ noninformative prior. *IEEE Transactions on Image Processing* 10(9), 1322–1331.
- Geweke, J. and M. Keane (2007). Smoothly mixing regressions. *Journal of Econometrics* 138(1), 252–290.
- Gruen, B. and F. Leisch (2023). *flexmix: Flexible Mixture Modeling*. R package version 2.3-19.
- Grün, B. and F. Leisch (2007). Fitting finite mixtures of generalized linear regressions in R. *Computational Statistics & Data Analysis* 51(11), 5247–5252.
- Grün, B. and F. Leisch (2008). FlexMix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software* 28(4), 1–35.
- Leisch, F. (2004). FlexMix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software* 11(8), 1–18.
- Mazza, A., A. Punzo, and S. Ingrassia (2018). flexCWM: A flexible framework for cluster-weighted models. *Journal of Statistical Software* 86(2), 1–30.
- Murphy, K. and T. B. Murphy (2020). Gaussian parsimonious clustering models with covariates and a noise component. *Advances in Data Analysis and Classification* 14(2), 293–325.
- Murphy, K. and T. B. Murphy (2023). *MoEClust: Gaussian Parsimonious Clustering Models with Covariates and a Noise Component*. R package version 1.5.2.
- Papastamoulis, P. (2016). label.switching: An R package for dealing with the label switching problem in MCMC outputs. *Journal of Statistical Software, Code Snippets* 69(1), 1–24.

- Papastamoulis, P. and G. Iliopoulos (2010). An artificial allocations based solution to the label switching problem in Bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics* 19(2), 313–331.
- Perrakis, K., T. Lartigue, F. Dondelinger, and S. Mukherjee (2023). Regularized joint mixture models. *Journal of Machine Learning Research* 24, 1–47.
- Städler, N. and S. Mukherjee (2013). Penalized estimation in high-dimensional hidden Markov models with state-specific graphical models. *The Annals of Applied Statistics* 7(4), 2157–2179.
- Wadsworth, I., M. L. Van Horn, and T. Jaki (2018). A diagnostic tool for checking assumptions of regression mixture models. *JP Journal of Biostatistics* 15(1), 1.